K 1

```
UUU         UUU  EEEEEEEEEEEEEEE   TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE   TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE   TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEE                     TTT        PPP         PPP
UUU         UUU  EEE                     TTT        PPP         PPP
UUU         UUU  EEE                     TTT        PPP         PPP
UUU         UUU  EEE                     TTT        PPP         PPP
UUU         UUU  EEE                     TTT        PPP         PPP
UUU         UUU  EEE                     TTT        PPP         PPP
UUU         UUU  EEEEEEEEEEEE            TTT        PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE            TTT        PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE            TTT        PPPPPPPPPPPP
UUU         UUU  EEE                     TTT        PPP
UUU         UUU  EEE                     TTT        PPP
UUU         UUU  EEE                     TTT        PPP
UUU         UUU  EEE                     TTT        PPP
UUU         UUU  EEE                     TTT        PPP
UUU         UUU  EEE                     TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE         TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE         TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE         TTT        PPP
```

```
**FILE**ID**RMSTEST4                                B 9
```

```
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT  44      44
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT  44      44
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT      44      44
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT      44      44
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT      44      44
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT      44      44
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT      4444444444
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT      4444444444
RR  RR      MM      MM          SS      TT      EE                  SS      TT              44
RR  RR      MM      MM          SS      TT      EE                  SS      TT              44
RR    RR    MM      MM          SS      TT      EE                  SS      TT              44      ....
RR    RR    MM      MM          SS      TT      EE                  SS      TT              44      ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT              44      ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT              44      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000            .IDENT  'V04-000'
0000            $BEGIN  RMSTEST4,009,__RMSTEST,<XAB RMS TEST PROGRAM>,<GBL,LONG>
0000
0000    ;
0000
0000            .ENABL  DBG
0000
0000    ;
0000    ; macros:
0000    ;
0000    ;
0000    ;
0000    ;
0000
0000            .MACRO  TYPE STRING, ?L
0000            STORE   <STRING>
0000            BLBC    VERBOSITY,L
0000            MOVL    #$$.TMPX,CMDORAB+RAB$L_RBF
0000            MOVW    #$$.TMPX1,CMDORAB+RAB$W_RSZ
0000            $PUT    RAB=CMDORAB,ERR=REPORT_ERROR
0000            BSBW    ERR
0000    L:
0000            .ENDM   TYPE
0000
0000    ;
0000
0000            .MACRO  STORE STRING,PRE
0000            .SAVE
0000            .PSECT  __$RMSNAM
0000            $$.TMPX=.
0000            PRE                                     ; store any carriage control info
0000            .ASCII  %STRING%
0000            $$.TMPX1=.-$$.TMPX
0000            .RESTORE
0000            .ENDM   STORE
```

```
0000     ;
0000
0000            .MACRO  BEGIN TSTNAM
0000            STORE   <TSTNAM>
0000            MOVL    #$$.TMPX,BEG_DESCR+4     ; addr
0000            MOVL    #$$.TMPX1,BEG_DESCR      ; len
0000            BSBW    BEGPUT
0000            .ENDM   BEGIN
0000            .MACRO  FINISH TSTNAM
0000            STORE   <TSTNAM>
0000            MOVL    #$$.TMPX,FIN_DESCR+4     ; addr
0000            MOVL    #$$.TMPX1,FIN_DESCR      ; len
0000            BSBW    FINPUT
0000            .ENDM   FINISH
0000            .MACRO  FIELD FLDNAM
0000            STORE   <FLDNAM>
0000            MOVL    #$$.TMPX,FLD_DESCR+4     ; addr
0000            MOVL    #$$.TMPX1,FLD_DESCR      ; len
0000            BSBW    FLDPUT
0000            .ENDM   FIELD
```

```
                          00000000          .PSECT   RMSTEST,GBL,LONG
                              0000          .ALIGN  LONG
                              0000  T4START::
                              0000  T4FAB:: $FAB     FNM=<TST$DISK:T4FILE.DAT;1>,-
                              0000                   ORG=SEQ,-
                              0000                   RFM=VFC,-
                              0000                   RAT=CR,-
                              0000                   FSZ=4,-
                              0000                   MRS=100,-
                              0000                   NAM=NAMBLK,-
                              0000                   DEQ=12
                              0050  FLUSH_FAB::
                              0050          $FAB     FAC=<PUT,GET>,-
                              0050                   FNM=<TST$DISK:T4FILE.DAT;1>,-
                              0050                   NAM=NAMBLK,-
                              0050                   SHR=<PUT,GET,UPI>,-
                              0050                   XAB=FHCXAB
                              00A0
                              00A0  ;
                              00A0  ;       attention: in order to assemble this module, t4rab and FLUSH_RAB
                              00A0  ;                  have been put into another module, RMSTESTR
                              00A0  ;
                              00A0
                              00A0  FHCXAB::
                              00A0          $XABFHC NXT=ALQXAB
                              00CC  ALQXAB::
                              00CC          $XABALL NXT=PROXAB,-
                              00CC                  DEQ=15
                              00EC  PROXAB::
                              00EC          $XABPRO
                              0144  DATXAB::
                              0144          $XABDAT
                              0170  RDTXAB::
                              0170          $XABRDT
                              0184  TRMXAB::
                              0184          $XABTRM
                              01A8          $RMSDEFEND
                  00000024    01A8                   EXTRA=XAB$L_SBN-4         ; 4 bytes of extra (spare) char.
                              01A8  SAVEPRO:
                      0000    01A8          .WORD   0                         ; word to save pro in
                              01AA
                              01AA  ;
                              01AA  ;THESE ARE THE DATA STRUCTURES FOR DATE AND TIME XAB CHECKS
                              01AA  ;
                              01AA
20 33 36 39 31 2D 52 41 4D 2D 33 20  01AA  CDT:    .ASCII  / 3-MAR-1963 03:03:03.03/
   33 30 2E 33 30 3A 33 30 3A 33 30  01B6
                          00000017  01C1           CDTL=.-CDT
20 34 34 39 31 2D 52 50 41 2D 34 20  01C1  RDT:    .ASCII  / 4-APR-1944 04:04:04.04/
   34 30 2E 34 30 3A 34 30 3A 34 30  01CD
                          00000017  01D8           RDTL=.-RDT
20 38 38 39 31 2D 47 55 41 2D 38 20  01D8  EDT:    .ASCII  / 8-AUG-1988 08:08:08.08/
   38 30 2E 38 30 3A 38 30 3A 38 30  01E4
                          00000017  01EF           EDTL=.-EDT
20 38 34 39 31 2D 43 45 44 2D 32 31  01EF  RDT2:   .ASCII  /12-DEC-1948 12:12:12.12/
   32 31 2E 32 31 3A 32 31 3A 32 31  01FB
                          00000017  0206           RDTL2=.-RDT2
```

```
000001AA'00000017  0206    188 CDTDEC: .LONG    CDTL,CDT
000001C1'00000017  020E    189 RDTDEC: .LONG    RDTL,RDT
000001D8'00000017  0216    190 EDTDEC: .LONG    EDTL,EDT
                   021E    191 RDT2DEC:
000001EF'00000017  021E    192         .LONG    RDTL2,RDT2
                   0226    193
             0000  0226    194 CURRVN: .WORD    0
             0000  0228    195 LEN:    .WORD    0                          ; length of returned string
         00000243  022A    196 CMPDAT: .BLKB    25                         ; has room for longest possible date
                   0243    197 CMPDATDEC:
0000022A'00000019  0243    198         .LONG    25,CMPDAT
         00000000  024B    199 CURRDT: .LONG    0                          ; address of current rdt string
         00000000  024F    200 UIC:    .LONG    0                          ; room to save current uic
         0000000B  0253    201              DATLEN=11                     ; length of date
         00000014  0253    202              TIMLEN=20                     ; length of ascii date and time
                   0253    203
                   0253    204
```

```
                                 0253  RMT$TEST_4A::
                          0FFC   0253          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                 0255          BEGIN   <XAB TESTS>
                                 026A
                                 026A  ;
                                 026A  ; create a file -- sys$disk:t4file.dat;1 -- with controlled attributes
                                 026A  ; and write 1 record to it, to further control the attributes
                                 026A  ;
                                 026A
       5B   FD92 CF        DE    026A          MOVAL   T4FAB,R11                ; r11 will be fab throughout
                                 026F                                           ; initialize values for restartability
                                 026F          $FAB_STORE      FAB=R11,-
                                 026F                  SHR=<PUT,GET,UPI>,-
                                 026F                  FOP=<SUP,CTG>,-
                                 026F                  XAB=FHCXAB,-
                                 026F                  ALQ=#0,-
                                 026F                  FAC=PUT
                                 0289
            FE1D CF        B4    0289          CLRW    XAB$W_LRL+FHCXAB         ; make sure lrl is 0
                                 028D          $XABPRO_STORE   XAB=PROXAB,-
                                 028D                  PRO=<RWED,RWED,RD,RWED>,-
                                 028D                  UIC=<333,44>
    A8 AF        0C A0     D0    02A2          MOVL    XAB$L_UIC(R0),UIC        ; save it for checking
  FEFB CF        08 A0     B0    02A7          MOVW    XAB$W_PRO(R0),SAVEPRO    ; ditto
                                 02AD          $XABALL_STORE   XAB=ALQXAB,-
                                 02AD                  ALQ=#10,-
                                 02AD                  AOP=CTG
                                 02BB          $CREATE FAB=R11,-                ; with all xabs linked in
                                 02BB                  ERR=REPORT_ERROR
               FD33'       30    02CA          BSBW    ERR
          00000000'8F      E1    02CD          BBC     #DEV$V_SQD,-
             03 40 AB            02D3                  FAB$L_DEV(R11),10$
                03F7       31    02D6          BRW     MTA                      ; if mta, skip this
       0A   FDFF CF        D1    02D9  10$:    CMPL    ALQXAB+XAB$L_ALQ,#10     ; allocated 10 blks?
                  15       18    02DE          BGEQ    RIGHT
                                 02E0          FIELD   <ALQ IN XAB ( NOT = DESIRED ALLOC ON CREATE)>
                                 02F5  RIGHT:
    15 04 AB      15       E1    02F5          BBC     #FAB$V_CBT,FAB$L_FOP(R11),OK1
                                 02FA          FIELD   <CBT BIT SET, THEREFORE>
    15 04 AB      14       E0    030F  OK1:    BBS     #FAB$V_CTG,FAB$L_FOP(R11),OK2
                                 0314          FIELD   <CTG BIT CLEAR, THEREFORE>
                                 0329  OK2:
                04FB       30    0329          BSBW    ZERO_XABS
                                 032C          $DISPLAY        FAB=R11,-
                                 032C                  ERR=REPORT_ERROR
               FCC2'       30    033B          BSBW    ERR
                0501       30    033E          BSBW    CHECK_CR                 ; check xabs
                                 0341          TYPE    <OK AFTER CREATE AND DISPLAY>
                                 0370
                                 0370  ;
                                 0370  ;extend
                                 0370  ;
                                 0370
    FD67 CF       30       D0    0370          MOVL    #48,XAB$L_ALQ+ALQXAB     ; set up
         FD57 CF           D4    0375          CLRL    XAB$L_NXT+ALQXAB
    24 AB   FD4F CF        DE    0379          MOVAL   ALQXAB,FAB$L_XAB(R11)
                                 037F          $EXTEND FAB=R11,-                ; extend file after create
```

```
                            037F                    ERR=REPORT_ERROR
                            038E
                            038E      ;
                            038E      ;using alq from xab
                            038E      ;
                            038E
                  FC6F'  30 038E            BSBW    ERR
         30   FD47 CF    D1 0391            CMPL    ALQXAB+XAB$L_ALQ,#48     ; alq in xab should ret actual alq
                    15   18 0396            BGEQ    ALQOK
                            0398            FIELD   <ALQ IN XAB (NOT = DESIRED ALLOC ON EXTEND)>
                            03AD      ALQOK:
                            03AD
                            03AD      ;
                            03AD      ;undo damage to xab links
                            03AD      ;
                            03AD
       24 AB   FCEF CF   DE 03AD            MOVAL   FHCXAB,FAB$L_XAB(R11)
     FCEA CF   FD15 CF   DE 03B3            MOVAL   ALQXAB,XAB$L_NXT+FHCXAB
     FD0F CF   FD2E CF   DE 03BA            MOVAL   PROXAB,XAB$L_NXT+ALQXAB
                            03C1            $CONNECT        RAB=T4RAB,-
                            03C1                    ERR=REPORT_ERROR
                  FC29'  30 03D4            BSBW    ERR
   32   41 8F    6E   00 2C 03D7            MOVC5   #0,(SP),#^A/A/,#50,CPYBUF; move 50 a's into cpybuf
            00000000'EF     03DD
00000000'EF  00000000'EF DE 03E2            MOVAL   CPYBUF,RAB$L_RBF+T4RAB
      00000000'EF   32   B0 03ED            MOVW    #50,RAB$W_RSZ+T4RAB
                            03F4            $PUT    RAB=T4RAB,-
                            03F4                    ERR=REPORT_ERROR
                  FBF6'  30 0407            BSBW    ERR
```

```
                                  040A
                                  040A  ;
                                  040A  ; take some time to try out flush
                                  040A  ;
                                  040A
                                  040A          $FLUSH  RAB=T4RAB,-
                                  040A                  ERR=REPORT_ERROR
                      FBE0'  30   041D          BSBW    ERR
                      0404   30   0420          BSBW    ZERO_XABS
                                  0423          $OPEN   FAB=FLUSH_FAB,-
                                  0423                  ERR=REPORT_ERROR
                      FBC9'  30   0434          BSBW    ERR
                      0418   30   0437          BSBW    CHECK_XABS
                                  043A          $CONNECT        RAB=FLUSH_RAB,-
                                  043A                  ERR=REPORT_ERROR
                      FBB0'  30   044D          BSBW    ERR
                                  0450          $GET    RAB=FLUSH_RAB,-
                                  0450                  ERR=REPORT_ERROR
                      FB9A'  30   0463          BSBW    ERR
           32   00000000'EF  B1   0466          CMPW    RAB$W_RSZ+FLUSH_RAB,#50 ; got right rec. size
                        15   13   046D          BEQL    RSZ_OK
                                  046F          FIELD   <RSZ IN RAB>
                                  0484  RSZ_OK:
00   41 8F  00000000'EF  32  2D   0484          CMPC5   #50,CPYBUF,#^A/A/,#0,(SP); is record ok?
                         6E       048E
                         15  13   048F          BEQL    YES
                                  0491          FIELD   <RECORD>
                                  04A6  YES:
                                  04A6          $GET    FLUSH_RAB               ; this should be eof
        00000000'8F  50  D1       04B3          CMPL    R0,#RMS$_EOF
                     0A  13       04BA          BEQL    10$
     5A  00000000'EF  DE          04BC          MOVAL   FLUSH_RAB,R10
                  FB3A'  30       04C3          BSBW    EOFPUT
                                  04C6  10$:
               FBAA CF   D4       04C6          CLRL    FAB$L_XAB+FLUSH_FAB     ; no xabs on close for now
                                  04CA          $DISCONNECT     RAB=FLUSH_RAB,- ; clean up after flush
                                  04CA                  ERR=REPORT_ERROR
                  FB20'  30       04DD          BSBW    ERR
                                  04E0          $CLOSE  FAB=FLUSH_FAB,-         ; all done w/ flush test
                                  04E0                  ERR=REPORT_ERROR
                  FB0C'  30       04F1          BSBW    ERR
                                  04F4          TYPE    <ALL DONE WITH FLUSH TEST>
                                  0523
                                  0523  ;
                                  0523  ; all done with flush test
                                  0523  ;
                                  0523
```

```
                          0523
                          0523
                          0523          $DISCONNECT     RAB=T4RAB,-
                          0523                          ERR=REPORT_ERROR
              FAC7'   30  0536          BSBW    ERR
                          0539          $CLOSE  FAB=R11,-
                          0539                  ERR=REPORT_ERROR
              FAB5'   30  0548          BSBW    ERR
                          054B
             17 AB    94  054B          CLRB    FAB$B_SHR(R11)
     24 AB   FB4E CF  DE  054E          MOVAL   FHCXAB,FAB$L_XAB(R11)   ; set up xab links again
              02D0    30  0554          BSBW    ZERO_XABS
                          0557          $OPEN   FAB=R11,-
                          0557                  ERR=REPORT_ERROR
              FA97'   30  0566          BSBW    ERR
                          0569
   15 04 AB    15     E1  0569          BBC     #FAB$V_CBT,FAB$L_FOP(R11),CC
                          056E          FIELD   <CBT BIT WAS SET, THEREFORE>
   15 04 AB    14     E1  0583  CC:     BBC     #FAB$V_CTG,FAB$L_FOP(R11),OK; after extend, not ctg
                          0588          FIELD   <CTG BIT WAS SET, THEREFORE>
                          059D
                          059D  ;
                          059D  ; check fhc xab
                          059D  ;
                          059D
              02B2    30  059D  OK:     BSBW    CHECK_XABS
                          05A0
                          05A0  ;
                          05A0  ;do another extend, forcing it to get the value from the alq of the fab
                          05A0  ;
                          05A0
             24 AB    D4  05A0          CLRL    FAB$L_XAB(R11)
        10 AB   0C    D0  05A3          MOVL    #12,FAB$L_ALQ(R11)
                          05A7          $EXTEND FAB=R11,-
                          05A7                  ERR=REPORT_ERROR
              FA47'   30  05B6          BSBW    ERR
        0C   10 AB    D1  05B9          CMPL    FAB$L_ALQ(R11),#12      ; alq in fab=12
                15    18  05BD          BGEQ    ALQOKT
                          05BF          FIELD   <ALQ IN FAB (NOT = DESIRED ALLOCATION AFTER EXTEND)>
                          05D4  ALQOK1:
                          05D4
                          05D4  ;
                          05D4  ;change protection and uic on close
                          05D4  ;
                          05D4
                          05D4          $XABPRO_STORE   XAB=PROXAB,-
                          05D4                  PRO=<RWED,RWED,RED,RWED>,-
                          05D4                  UIC=<222,55>
        24 AB   50    D0  05E9          MOVL    R0,FAB$L_XAB(R11)               ; set up xab
   FBB5 CF   08 A0    B0  05ED          MOVW    XAB$W_PRO(R0),SAVEPRO           ; for check
   FC56 CF   0C A0    D0  05F3          MOVL    XAB$L_UIC(R0),UIC               ; ditto
                          05F9          $CLOSE  FAB=R11,-
                          05F9                  ERR=REPORT_ERROR
              F9F5'   30  0608          BSBW    ERR
     24 AB   FA91 CF  DE  060B          MOVAL   FHCXAB,FAB$L_XAB(R11)
                          0611          $OPEN   FAB=R11,-                       ; check changes after ext
                          0611                  ERR=REPORT_ERROR
```

```
                    F9DD'  30  0620             BSBW    ERR
        15 04 AB     15    E1  0623             BBC     #FAB$V_CBT,FAB$L_FOP(R11),NOCBT
                               0628             FIELD   <CBT BIT WAS SET, THEREFORE>
        15 04 AB     14    E1  063D   NOCBT:    BBC     #FAB$V_CTG,FAB$L_FOP(R11),NOCTG ; shouldn't be ctg, after extend
                               0642             FIELD   <CTG BIT WAS SET, AFTER 2 EXTENDS, THEREFORE>
                               0657   NOCTG:
             59   FA45 CF  DE  0657             MOVAL   FHCXAB,R9                       ; check pertinent fields
    00000046 8F     0C A9  D1  065C             CMPL    XAB$L_HBK(R9),#70               ; alq=10+48+12
                       15  18  0664             BGEQ    HBKOK
                               0666             FIELD   <HBK IN FHCXAB (AFTER 2ND EXTEND)>
             00     28 A9  D1  067B   HBKOK:    CMPL    XAB$L_SBN(R9),#0                ; not ctg anymore
                       15  13  067F             BEQL    STILL_OK
                               0681             FIELD   <SBN IN FHCXAB (AFTER 2ND EXTEND)>
                               0696   STILL_OK:
                     01C9  30  0696             BSBW    CHECK_ALL
                    00 46      0699             .BYTE   70,0                            ; values for alq,ctg (not ctg)
                     025E  30  069B             BSBW    CHECK_PRO
                               069E             TYPE    <DONE WITH 2ND EXTEND, NOW TEST DATES>
                    24 AB  D4  06CD             CLRL    FAB$L_XAB(R11)                  ; no xabs on this close, for now
                               06D0   MTA:      $CLOSE  FAB=R11,-                       ; continue if mta
                               06D0                     ERR=REPORT_ERROR
                    F91E'  30  06DF             BSBW    ERR
                               06E2
                               06E2   ;
                               06E2   ;before finishing up, have some fun with the dat and rdt xabs
                               06E2   ;
                               06E2
             59   FA5E CF  DE  06E2             MOVAL   DATXAB,R9
                               06E7             $BINTIM_S       CDTDEC,XAB$Q_CDT(R9)
                               06F5             $BINTIM_S       RDTDEC,XAB$Q_RDT(R9)
                               0703             $BINTIM_S       EDTDEC,XAB$Q_EDT(R9)
        08 A9     00C8 8F  B0  0711             MOVW    #200,XAB$W_RVN(R9)
      FB08 CF     00C8 8F  B0  0717             MOVW    #200,CURRVN
           24 AB       59  D0  071E             MOVL    R9,FAB$L_XAB(R11)
                    04 A9  D4  0722             CLRL    XAB$L_NXT(R9)
      FB1F CF     FA98 CF  DE  0725             MOVAL   RDT,CURRDT                      ; current rdt str
    04 AB     00000080 8F  C8  072C             BISL    #FAB$M_RWO,FAB$L_FOP(R11)       ; rewind if mta
                               0734             $CREATE FAB=R11,-
                               0734                     ERR=REPORT_ERROR
                    F8BA'  30  0743             BSBW    ERR
      F9FB CF     FA26 CF  DE  0746             MOVAL   RDTXAB,XAB$L_NXT+DATXAB
                     00C6  30  074D             BSBW    ZERO_DAT_XABS
                               0750             $DISPLAY        FAB=R11,-
                               0750                     ERR=REPORT_ERROR
                    F89E'  30  075F             BSBW    ERR
                     032B  30  0762             BSBW    CHECK_DATES
      FADF CF     FA86 CF  DE  0765             MOVAL   RDT2,CURRDT                     ; get a new rdt
           59     FA00 CF  DE  076C             MOVAL   RDTXAB,R9
                               0771             $BINTIM_S       RDT2DEC,XAB$Q_RDT(R9)
        08 A9     012C 8F  B0  077F             MOVW    #300,XAB$W_RVN(R9)
      FA9A CF     012C 8F  B0  0785             MOVW    #300,CURRVN
           24 AB       59  D0  078C             MOVL    R9,FAB$L_XAB(R11)               ; only rdt for close
                               0790             $CLOSE  FAB=R11,-
                               0790                     ERR=REPORT_ERROR
                    F85E'  30  079F             BSBW    ERR
        24 AB     F99E CF  DE  07A2             MOVAL   DATXAB,FAB$L_XAB(R11)           ; only dat for open
                  F99C CF  D4  07A8             CLRL    XAB$L_NXT+DATXAB
```

```
                 68   10  07AC        BSBB    ZERO_DAT_XABS
                          07AE        $FAB_STORE      FAB=R11,-            ; can't "put" to mta
                          07AE                FAC=GET
                          07B2        $OPEN   FAB=R11,-
                          07B2                ERR=REPORT_ERROR
               F83C'  30  07C1        BSBW    ERR
       24 AB  F9AB CF DE  07C4        MOVAL   RDTXAB,FAB$L_XAB(R11)        ; get rdt on display
                          07CA        $DISPLAY        FAB=R11,-
                          07CA                ERR=REPORT_ERROR
               F824'  30  07D9        BSBW    ERR
                02B1  30  07DC        BSBW    CHECK_DATES
               24 AB  D4  07DF        CLRL    FAB$L_XAB(R11)               ; no xabs for this close
                          07E2        $CLOSE  FAB=R11,-
                          07E2                ERR=REPORT_ERROR
               F80C'  30  07F1        BSBW    ERR
                          07F4
                          07F4
                          07F4        $FAB_STORE      FAB=R11,-            ; restore fac
                          07F4                FAC=PUT
  04 AB   00000080 8F CA  07F8        BICL    #FAB$M_RWO,FAB$L_FOP(R11)    ; and fop
                          0800        FINISH  <XAB TESTS>
                      04  0815        RET
```

```
                                             0816
                                             0816    ;
                                             0816    ;2 routines to zero out the xabs before checking the results
                                             0816    ;
                                             0816
                                             0816    ZERO_DAT_XABS:
                                             0816
                                             0816    ;
                                             0816    ;zero out the dat and rdt xabs
                                             0816    ;
                                             0816
  F92C CF    26    00    6E    00    2C    0816              MOVC5   #0,(SP),#0,#<XAB$C_DATLEN-6>,DATXAB+6
  F950 CF    0E    00    6E    00    2C    081E              MOVC5   #0,(SP),#0,#<XAB$C_RDTLEN-6>,RDTXAB+6
                                     05    0826              RSB
                                             0827
                                             0827    ZERO_XABS:
                                             0827
                                             0827    ;
                                             0827    ;zero out the fhc, all and pro xabs
                                             0827    ;
                                             0827
  F877 CF    26    00    6E    00    2C    0827              MOVC5   #0,(SP),#0,#<XAB$C_FHCLEN-6>,FHCXAB+6
  F89B CF    1A    00    6E    00    2C    082F              MOVC5   #0,(SP),#0,#<XAB$C_ALLLEN-6>,ALQXAB+6
       0052 8F    00    6E    00    2C    0837              MOVC5   #0,(SP),#0,#<XAB$C_PROLEN-6>,PROXAB+6
                    F8B1 CF                083E
                                     05    0841              RSB
```

```
                        0842   CHECK_CR:
                        0842
                        0842   ;
                        0842   ;routine to check xabs after create and subsequent displ.
                        0842   ;
                        0842
             00F7  30   0842           BSBW    CHECK_FHC
         01 00 0A  00   0845           .BYTE   0,10,0,1                ; values for lrl,alq,ffb,sbn
             0016  30   0849           BSBW    CHECK_ALL
               01  0A   084C           .BYTE   10,1                    ; values for alq,ctg ( should be)
             00AB  30   084E           BSBW    CHECK_PRO
                   05   0851           RSB
                        0852
                        0852   CHECK_XABS:
                        0852
                        0852   ;
                        0852   ;general routine to check out all xabs
                        0852   ;
                        0852
             00E7  30   0852           BSBW    CHECK_FHC
         00 38 3A  32   0855           .BYTE   50,58,56,0              ; values for lrl,alq,ffb,sbn(not ctg)
             0006  30   0859           BSBW    CHECK_ALL
               00  3A   085C           .BYTE   58,0                    ; values for alq,ctg ( not ctg anymore)
             009B  30   085E           BSBW    CHECK_PRO
                   05   0861           RSB
                        0862
                        0862   CHECK_ALL:
                        0862
                        0862   ;
                        0862   ;routine to check out the allocation xab
                        0862   ;
                        0862
   59    F866 CF   DE   0862           MOVAL   ALQXAB,R9
     0F    14 A9   B1   0867           CMPW    XAB$W_DEQ(R9),#15
             15    13   086B           BEQL    DEQOK
                        086D           FIELD   <DEQ IN ALL. XAB>
     00    16 A9   91   0882   DEQOK:  CMPB    XAB$B_BKZ(R9),#0
             15    13   0886           BEQL    BKZOK
                        0888           FIELD   <BKZ IN ALL. XAB>
     50    00 BE   9A   089D   BKZOK:  MOVZBL  @(SP),R0
              6E   D6   08A1           INCL    (SP)
     10 A9    50   D1   08A3           CMPL    R0,XAB$L_ALQ(R9)
              15   15   08A7           BLEQ    ALQOK2
                        08A9           FIELD   <ALQ IN ALL. XAB>
     50    00 BE   9A   08BE   ALQOK2: MOVZBL  @(SP),R0
              6E   D6   08C2           INCL    (SP)
          1A  50   E9   08C4           BLBC    R0,NOTCTG
  2F 08 A9    07   E0   08C7           BBS     #XAB$V_CTG,XAB$B_AOP(R9),AOPOK  ; should be set
                        08CC           FIELD   <CTG CLR IN AOP, THEREFORE>
  15 08 A9    07   E1   08E1   NOTCTG: BBC     #XAB$V_CTG,XAB$B_AOP(R9),AOPOK  ; should be clear
                        08E6           FIELD   <CTG SET IN AOP, THEREFORE>
                   05   08FB   AOPOK:  RSB
                        08FC
                        08FC   CHECK_PRO:
                        08FC
                        08FC   ;
                        08FC   ;check the protection xab
```

```
                                        08FC     559 ;
                                        08FC     560
            59    F7EC CF        DE     08FC     561          MOVAL   PROXAB,R9
         08 A9    F8A3 CF        B1     0901     562          CMPW    SAVEPRO,XAB$W_PRO(R9)    ; cmp to saved value
                          15     13     0907     563          BEQL    PROOK
                                        0909     564          FIELD   <PROT FIELD IN PROT XAB>
         0C A9    F92D CF        D1     091E     565 PROOK:   CMPL    UIC,XAB$L_UIC(R9)
                          15     13     0924     566          BEQL    UICOK
                                        0926     567          FIELD   <UIC FIELD IN PROT. XAB>
                                 05     093B     568 UICOK:   RSB
                                        093C     569
```

```
                                  093C      CHECK_FHC:
                                  093C
                                  093C      ;
                                  093C      ;  check fhc xab carefully
                                  093C      ;
                                  093C
            59   F760 CF     DE   093C                MOVAL    FHCXAB,R9                  ; r9 is ptr to xab thru-out cmp's
                                  0941      EXTC:
             03    08 A9     91   0941                CMPB     XAB$B_RFO(R9),#FAB$C_VFC; check rec. format & org.
                      15     13   0945                BEQL     RFOC
                                  0947                FIELD    <RFO IN FHC XAB>
             09 A9    02     93   095C      RFOC:     BITB     #FAB$M_CR,XAB$B_ATR(R9)  ; check rat field
                      15     12   0960                BNEQ     ATRC
                                  0962                FIELD    <ATR IN FHC XAB>
             50    00 BE     9A   0977      ATRC:     MOVZBL   @(SP),R0
                      6E     D6   097B                INCL     (SP)
             50    0A A9     B1   097D                CMPW     XAB$W_LRL(R9),R0           ; check longest record len
                      15     13   0981                BEQL     LRLC
                                  0983                FIELD    <LRL IN FHC XAB>
             50    00 BE     9A   0998      LRLC:     MOVZBL   @(SP),R0
                      6E     D6   099C                INCL     (SP)
             50    0C A9     D1   099E                CMPL     XAB$L_HBK(R9),R0           ; check alq
                      15     18   09A2                BGEQ     HBKC
                                  09A4                FIELD    <HBK IN FHC XAB>
             01    10 A9     D1   09B9      HBKC:     CMPL     XAB$L_EBK(R9),#1           ; check end block
                      15     13   09BD                BEQL     EBKC
                                  09BF                FIELD    <EBK IN FHC XAB>
             50    00 BE     9A   09D4      EBKC:     MOVZBL   @(SP),R0
                      6E     D6   09D8                INCL     (SP)
             50    14 A9     B1   09DA                CMPW     XAB$W_FFB(R9),R0           ; check first free byte
                      15     13   09DE                BEQL     FFBC                       ; its len of rec + fsz + 2
                                  09E0                FIELD    <FFB IN FHC XAB>
             00    16 A9     91   09F5      FFBC:     CMPB     XAB$B_BKZ(R9),#0           ; check bucket size
                      15     13   09F9                BEQL     BKZC
                                  09FB                FIELD    <BKZ IN FHC XAB>
             04    17 A9     91   0A10      BKZC:     CMPB     XAB$B_HSZ(R9),#4           ; check fixed area size
                      15     13   0A14                BEQL     HSZC
                                  0A16                FIELD    <HSZ IN FHC XAB>
        0064 8F    18 A9     B1   0A2B      HSZC:     CMPW     XAB$W_MRZ(R9),#100         ; check max. rec size
                      15     13   0A31                BEQL     MRZC
                                  0A33                FIELD    <MRZ IN FHC XAB>
             0F    1A A9     B1   0A48      MRZC:     CMPW     XAB$W_DXQ(R9),#15          ; check def ext. qty
                      15     13   0A4C                BEQL     DXQC
                                  0A4E                FIELD    <DXQ IN FHC XAB>
             50    00 BE     9A   0A63      DXQC:     MOVZBL   @(SP),R0
                      6E     D6   0A67                INCL     (SP)
                   08 50     E9   0A69                BLBC     R0,10$
             00    28 A9     D1   0A6C                CMPL     XAB$L_SBN(R9),#0           ; make sure non-zero lbn
                                  0A70
                                  0A70      ;
                                  0A70      ;since it's ctg
                                  0A70      ;
                                  0A70
                      1D     12   0A70                BNEQ     FHC_OK
                      06     11   0A72                BRB      20$
             00    28 A9     D1   0A74      10$:      CMPL     XAB$L_SBN(R9),#0           ; make sure zero lbn
                      15     13   0A78                BEQL     FHC_OK                     ; since it isn't contig.
```

```
         0A7A    628 20$:     FIELD    <SBN IN FHC XAB>
    05   0A8F    629 FHC_OK: RSB
```

```
                                   0A90
                                   0A90      CHECK_DATES:
                                   0A90
                                   0A90      ;
                                   0A90      ;routine to check edt and cdt in dat xab
                                   0A90      ;and rdt and rvn in both dat and rdt xab's
                                   0A90      ;
                                   0A90
              F793 CF    0B    B0  0A90                 MOVW    #DATLEN,LEN                       ; default is check date only
     05 40 AB  00000000'8F     E0  0A95                 BBS     #DEV$V_SQD,FAB$L_DEV(R11),10$
              F785 CF    14    B0  0A9E                 MOVW    #TIMLEN,LEN                       ; if not mta, check date and time
              59   F69D CF     DE  0AA3      10$:       MOVAL   DATXAB,R9
                                   0AA8                 $ASCTIM_S       ,CMPDATDEC,XAB$Q_CDT(R9)
  F766 CF   F6E9 CF   F76A CF  29  0ABA                 CMPC3   LEN,CDT,CMPDAT
                            15 13  0AC4                 BEQL    CDTOK
                                   0AC6                 FIELD   <CDT IN DAT XAB>
                                   0ADB      CDTOK:
                                   0ADB                 $ASCTIM_S       ,CMPDATDEC,XAB$Q_EDT(R9)
     F735 CF   F6E6 CF    0B   29  0AED                 CMPC3   #DATLEN,EDT,CMPDAT                ; only check date
                            15 13  0AF5                 BEQL    EDTOK
                                   0AF7                 FIELD   <EDT IN DAT XAB>
                                   0B0C      EDTOK:
     01 40 AB   00000000'8F    E1  0B0C                 BBC     #DEV$V_SQD,FAB$L_DEV(R11),10$
                               05  0B15                 RSB
                                   0B16      10$:       $ASCTIM_S       ,CMPDATDEC,XAB$Q_RDT(R9)  ; that's it if mta
  F6F8 CF   F71C DF   F6FC CF  29  0B28                 CMPC3   LEN,@CURRDT,CMPDAT
                            15 13  0B32                 BEQL    RDTOK
                                   0B34                 FIELD   <RDT IN DAT XAB>
                                   0B49      RDTOK:
                                   0B49                 $ASCTIM_S       ,CMPDATDEC,XAB$Q_RDT+RDTXAB
  F6C4 CF   F6E8 DF   F6C8 CF  29  0B5C                 CMPC3   LEN,@CURRDT,CMPDAT
                            15 13  0B66                 BEQL    RDTOK1
                                   0B68                 FIELD   <RDT IN RDT XAB>
                                   0B7D      RDTOK1:
            08 A9   F6A5 CF    B1  0B7D                 CMPW    CURRVN,XAB$W_RVN(R9)
                            15 13  0B83                 BEQL    RVNOK
                                   0B85                 FIELD   <RVN IN DAT XAB>
          F5D7 CF   F688 CF    B1  0B9A      RVNOK:     CMPW    CURRVN,XAB$W_RVN+RDTXAB
                            15 13  0BA1                 BEQL    RVNOK1
                                   0BA3                 FIELD   <RVN IN RDT XAB>
                               05  0BB8      RVNOK1:    RSB
                                   0BB9                 .END
```

```
                                                          F 10
RMSTEST4                        XAB RMS TEST PROGRAM             16-SEP-1984 01:47:44  VAX/VMS Macro V04-00        Page  17
Symbol table                                                      5-SEP-1984 04:21:52  [UETP.SRC]RMSTEST4.MAR;1         (15)

$$.PSECT_EP          = 00000000              ERR                         ********    X  01
$$.TAB               = 00000184 R  D  01     EXTC                        00000941 R  D  01
$$.TABEND            = 000001A8 R  D  01     EXTRA                     = 00000024    D
$$.TMP               = 00000001    D         FAB$B_FAC                 = 00000016    D
$$.TMP1              = 00000002    D         FAB$B_FNS                 = 00000034    D
$$.TMP2              = 0000005B    D         FAB$B_SHR                 = 00000017    D
$$.TMP5              = 00000002    D         FAB$C_BID                 = 00000003    D
$$.TMPX              = 00000352 R  D  04     FAB$C_BLN                 = 00000050    D
$$.TMPX1             = 0000000E    D         FAB$C_SEQ                 = 00000000    D
$$RMSTEST            = 0000001E              FAB$C_VAR                 = 00000002    D
$$RMS_PBUGCHK        = 00000010              FAB$C_VFC                 = 00000003    D
$$RMS_TBUGCHK        = 00000008              FAB$L_ALQ                 = 00000010    D
$$RMS_UMODE          = 00000004              FAB$L_DEV                 = 00000040    D
..AFLG               = 00000000    D         FAB$L_FNA                 = 0000002C    D
..FLG                = 00000002    D         FAB$L_FOP                 = 00000004    D
..MOD                = 00000001    D         FAB$L_XAB                 = 00000024    D
..N                  = 00000001              FAB$M_CR                  = 00000002    D
..TYP                = 00000003    D         FAB$M_RWO                 = 00000080    D
.LEN                 = 00000001    D         FAB$V_CBT                 = 00000015    D
ALQOK                  000003AD R  D  01     FAB$V_CHAN_MODE           = 00000002    D
ALQOK1                 000005D4 R  D  01     FAB$V_CR                  = 00000001    D
ALQOK2                 000008BE R  D  01     FAB$V_CTG                 = 00000014    D
ALQXAB                 000000CC RG D  01     FAB$V_FILE_MODE           = 00000004    D
AOPOK                  000008FB R  D  01     FAB$V_GET                 = 00000001    D
ATRC                   00000977 R  D  01     FAB$V_LNM_MODE            = 00000000    D
BEGPUT                 ********    X  01     FAB$V_PUT                 = 00000000    D
BEG_DESCR              ********    X  01     FAB$V_SUP                 = 00000002    D
BKZC                   00000A10 R  D  01     FAB$V_UPI                 = 00000006    D
BKZOK                  0000089D R  D  01     FAB$W_GBC                 = 00000048    D
CC                     00000583 R  D  01     FFBC                        000009F5 R  D  01
CDT                    000001AA R  D  01     FHCXAB                      000000A0 RG D  01
CDTDEC                 00000206 R  D  01     FHC_OK                      00000A8F R  D  01
CDTL                 = 00000017    D         FINPUT                      ********    X  01
CDTOK                  00000ADB R  D  01     FIN_DESCR                   ********    X  01
CHECK_ALL              00000862 R  D  01     FLDPUT                      ********    X  01
CHECK_CR               00000842 R  D  01     FLD_DESCR                   ********    X  01
CHECK_DATES            00000A90 R  D  01     FLUSH_FAB                   00000050 RG D  01
CHECK_FHC              0000093C R  D  01     FLUSH_RAB                   ********    X  01
CHECK_PRO              000008FC R  D  01     HBKC                        000009B9 R  D  01
CHECK_XABS             00000852 R  D  01     HBKOK                       0000067B R  D  01
CMDORAB                ********    X  01     HSZC                        00000A2B R  D  01
CMPDAT                 0000022A R  D  01     LEN                         00000228 R  D  01
CMPDATDEC              00000243 R  D  01     LRLC                        00000998 R  D  01
CPYBUF                 ********    X  01     MRZC                        00000A48 R  D  01
CURRDT                 0000024B R  D  01     MTA                         000006D0 R  D  01
CURRVN                 00000226 R  D  01     NAMBLK                      ********    X  01
DATLEN               = 0000000B    D         NOCBT                       0000063D R  D  01
DATXAB                 00000144 RG D  01     NOCTG                       00000657 R  D  01
DEQOK                  00000882 R  D  01     NOTCTG                      000008E1 R  D  01
DEV$V_SQD              ********    X  01     OK                          0000059D R  D  01
DXQC                   00000A63 R  D  01     OK1                         0000030F R  D  01
EBKC                   000009D4 R  D  01     OK2                         00000329 R  D  01
EDT                    000001D8 R  D  01     PROOK                       0000091E R  D  01
EDTDEC                 00000216 R  D  01     PROXAB                      000000EC RG D  01
EDTL                 = 00000017    D         RAB$L_RBF                   ********    X  01
EDTOK                  00000B0C R  D  01     RAB$W_RSZ                   ********    X  01
EOFPUT                 ********    X  01     RDT                         000001C1 R  D  01
```

```
RDT2                             000001EF R   D   01
RDT2DEC                          0000021E R   D   01
RDTDEC                           0000020E R   D   01
RDTL                           = 00000017     D
RDTL2                          = 00000017     D
RDTOK                            00000B49 R   D   01
RDTOK1                           00000B7D R   D   01
RDTXAB                           00000170 RG  D   01
REPORT_ERROR                     ********     X   01
RFOC                             0000095C R   D   01
RIGHT                            000002F5 R   D   01
RMS$_EOF                         ********     X   01
RMT$TEST_4A                      00000253 RG  D   01
RSZ_OK                           00000484 R   D   01
RVNOK                            00000B9A R   D   01
RVNOK1                           00000BB8 R   D   01
SAVEPRO                          000001A8 R   D   01
STILL_OK                         00000696 R   D   01
SYS$ASCTIM                       ********    GX   01
SYS$BINTIM                       ********    GX   01
SYS$CLOSE                        ********    GX   01
SYS$CONNECT                      ********    GX   01
SYS$CREATE                       ********    GX   01
SYS$DISCONNECT                   ********    GX   01
SYS$DISPLAY                      ********    GX   01
SYS$EXTEND                       ********    GX   01
SYS$FLUSH                        ********    GX   01
SYS$GET                          ********    GX   01
SYS$OPEN                         ********    GX   01
SYS$PUT                          ********    GX   01
T4FAB                            00000000 RG  D   01
T4RAB                            ********     X   01
T4START                          00000000 RG  D   01
TIMLEN                         = 00000014     D
TRMXAB                           00000184 RG  D   01
UIC                              0000024F R   D   01
UICOK                            0000093B R   D   01
VERBOSITY                        ********     X   01
XAB$B_AID                      = 00000017     D
XAB$B_AOP                      = 00000008     D
XAB$B_ATR                      = 00000009     D
XAB$B_BKZ                      = 00000016     D
XAB$B_HSZ                      = 00000017     D
XAB$B_MTACC                    = 0000000A     D
XAB$B_PROT_MODE                = 00000010     D
XAB$B_PROT_OPT                 = 0000000B     D
XAB$B_RFO                      = 00000008     D
XAB$C_ALL                      = 00000014     D
XAB$C_ALLLEN                   = 00000020     D
XAB$C_DAT                      = 00000012     D
XAB$C_DATLEN                   = 0000002C     D
XAB$C_FHC                      = 0000001D     D
XAB$C_FHCLEN                   = 0000002C     D
XAB$C_PRO                      = 00000013     D
XAB$C_PROLEN                   = 00000058     D
XAB$C_RDT                      = 0000001E     D
XAB$C_RDTLEN                   = 00000014     D
XAB$C_TRM                      = 0000001F     D
XAB$C_TRMLEN                   = 00000024     D
XAB$L_ACLBUF                   = 00000018     D
XAB$L_ACLCTX                   = 00000020     D
XAB$L_ALQ                      = 00000010     D
XAB$L_EBK                      = 00000010     D
XAB$L_HBK                      = 0000000C     D
XAB$L_ITMLST                   = 00000008     D
XAB$L_LOC                      = 0000000C     D
XAB$L_NXT                      = 00000004     D
XAB$L_SBN                      = 00000028     D
XAB$L_UIC                      = 0000000C     D
XAB$Q_CDT                      = 00000014     D
XAB$Q_EDT                      = 0000001C     D
XAB$Q_RDT                      = 0000000C     D
XAB$V_CTG                      = 00000007     D
XAB$W_ACLSIZ                   = 0000001C     D
XAB$W_DEQ                      = 00000014     D
XAB$W_DXQ                      = 0000001A     D
XAB$W_FFB                      = 00000014     D
XAB$W_GRP                      = 0000000E     D
XAB$W_ITMLST_LEN               = 0000000C     D
XAB$W_LRL                      = 0000000A     D
XAB$W_MBM                      = 0000000C     D
XAB$W_MRZ                      = 00000018     D
XAB$W_PRO                      = 00000008     D
XAB$W_RFI0                     = 00000018     D
XAB$W_RFI2                     = 0000001A     D
XAB$W_RFI4                     = 0000001C     D
XAB$W_RVN                      = 00000008     D
XAB$W_VOL                      = 0000000A     D
YES                              000004A6 R   D   01
ZERO_DAT_XABS                    00000816 R   D   01
ZERO_XABS                        00000827 R   D   01
```

```
                                             +----------------+
                                             ! Psect synopsis !
                                             +----------------+

PSECT name                  Allocation           PSECT No.  Attributes
----------                  ----------           ---------  ----------
.  ABS  .                   00000000  (    0.)   00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
_RMSTEST                    00000BB9  ( 3001.)   01 (  1.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD     WRT NOVEC LONG
$ABS$                       00000000  (    0.)   02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$RMSNAM                     0000002A  (   42.)   03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
__$RMSNAM                   00000360  (  864.)   04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                             +------------------------+
                                             ! Performance indicators !
                                             +------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                    29    00:00:00.09    00:00:00.43
Command processing               105    00:00:00.63    00:00:02.75
Pass 1                           362    00:00:17.15    00:00:38.17
Symbol table sort                  0    00:00:00.53    00:00:01.16
Pass 2                           148    00:00:03.68    00:00:08.25
Symbol table output               24    00:00:00.15    00:00:00.31
Psect synopsis output              3    00:00:00.02    00:00:00.30
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals             673    00:00:22.25    00:00:51.37

The working set limit was 1350 pages.
80832 bytes (158 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 460 non-local and 9 local symbols.
672 source lines were read in Pass 1, producing 53 object records in Pass 2.
67 pages of virtual memory were used to define 50 macros.

                                             +--------------------------+
                                             ! Macro library statistics !
                                             +--------------------------+

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                            0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                        50
TOTALS (all libraries)                                   50

1077 GETS were required to define 50 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSTEST4/OBJ=OBJ$:RMSTEST4 MSRC$:RMSTEST4/UPDATE=(ENH$:RMSTEST4)+EXECML$/LIB
```

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RMSTEST4
LIS

RMSTEST7
LIS

RMSTEST8
LIS

RMSTESTR
LIS

RMSTEST0
LIS

RMSTEST2
LIS

RMSTEST3
LIS

RMSTEST5
LIS

RMSTEST6
LIS

RMSTESTT
LIS

RMSTEST1
LIS

SATSSF17
LIS